on the clients **116** by specifying the explicit metrics and/or defining functions for computed metrics and upload the templates to the template storage module **312**. Thus, the template storage module **312** stores a variety of templates, some created by the operator of the metrics server **112** and others created by end-users and/or other entities.

[0039] In one embodiment, an ad server module **314** serves advertisements and/or other messages to clients. The ads are stored in the ad serving module **314** and/or at another location on the network **114**. In one embodiment, the ad server module **314** selects ads based on a template being used by an end-user and/or the metrics presented by the template. For example, if an end-user selects a "house hunting" template, the ad server module **314** serves ads related to real estate, such as ads for real estate agents, mortgage brokers, etc. In another example, if the metrics presented by a template selected by an end-user show average incomes, the ad serving module **314** shows ads for more expensive products when the metrics indicate a high average income and for less expensive products when the metrics indicate a lower average income. Other embodiments of the ad serving module **314** serve ads based on additional and/or different criteria.

[0040] A communications module **316** supports communications with the map server **110**, clients **116**, and/or other entities via the network **114**. In one embodiment, the communications module **316** receives communications from the clients **116** including requests for templates and/or requests for metrics data. In response, the communications module **316** provides the templates, data, and/or ads to the requesting clients **116**. Similarly, an embodiment of the communications module **316** receives new templates and/or metrics data from the clients **116** and/or other entities and provides these inputs to other modules in the metrics server **112** for processing and/or storage.

[0041] FIG. **4** is a high-level block diagram illustrating modules within the mapping engine **118** according to one embodiment. Other embodiments have different and/or additional modules than the ones shown in the figure. In addition, other embodiments distribute the functionalities among the modules in a different manner. Further, in some embodiments functionalities attributed to the mapping engine **118** are performed by other entities on the client **116** and/or elsewhere on the network **114**.

[0042] In one embodiment, the mapping engine **118** is a standalone application. In another embodiment, the mapping engine **118** is incorporated into another application. For example, the modules can be executed within an execution environment provided by a general purpose web browser.

[0043] A communications module **410** supports communications with the map server **110**, metrics server **112**, and/or other entities via the network **114**. In one embodiment, the communications module **410** uses Asynchronous JavaScript and XML (AJAX) to communicate with the map server **110** and/or metrics server **112**. This technology allows the communications module **410** to rapidly exchange small amounts of data with the servers and increases the perceived responsiveness of the mapping engine **118** when updating maps, displaying metrics, and/or performing other functions.

[0044] In one embodiment, the communications module **410** submits requests to the map server **110** for map data and receives map data in response. Likewise, the communications module **410** submits requests for templates and/or metrics data from the metrics server **112** and receives the requested data in response. The communications module **410** provides received data to the other modules in the mapping engine **118** for subsequent processing. In some embodiments, the communications module **410** provides templates and/or metrics data to the metrics server **112**.

[0045] A map module **412** requests, receives, and stores map data from the map server **110**. The map module **412** provides the map server **110** with a description of a geographic region, such as the name of a city, a latitude/longitude on which the map is centered, a set of coordinates defining a region for which map data are desired, etc. and the map server **110** provides the requested map data. These data are stored by the map module **412** for use by the other modules in the mapping engine **118**. In one embodiment, the map module **412** caches the map data for a specified period of time, after which the map data are discarded.

[0046] A template module **414** requests, receives, and stores templates. The templates are requested and received from the metrics server **112**, another server on the network **114** and/or created by an end-user of the client **116**. In one embodiment, one or more templates are designated as "active" by the end-user, meaning that the metrics specified by the template are displayed in combination with a map at the client **116**.

[0047] A metric storage module **416** stores metrics data received from the metrics server **112** and/or provided by the end-user. In one embodiment, an end-user can provide metrics data and selectively override data received from the metrics server **112**. For example, if the end-user is aware that a noise level at a particular location is higher than the data from the metrics server **112** indicate, the end-user can enter a new noise level. The metric storage module **416** stores the user-provided data. In some embodiments, the mapping engine **118** provides the end-user provided data to the metrics server **112**, from where it can be shared with other clients **116**.

[0048] A metric computation module **418** determines values of any computed metrics specified and/or required for a template. In one embodiment, the metric computation module **418** operates in real time. Thus, if a computed metric in an active template is the distance between a fixed location and a variable location, the metric computation module **418** updates the metric in real time as the variable location changes. In one embodiment, the metric computation module **418** searches map and/or metric data to find locations where metrics have certain values. For example, the metric computation module **418** can identify areas within five minutes walking distance to schools that have average home prices below $500K.

[0049] A user interface (UI) module **420** enables and controls end-user interactions with the mapping engine **418**. In one embodiment, the UI module **420** interacts with the operating system and/or other modules on the client **116** to provide a graphical user interface (GUI) that allows an end-user to use input devices such as keyboards and mice to perform actions such as selecting areas to map, scrolling around the map, and controlling the level of detail (e.g., zoom) of the map. In addition, the GUI allows the end-user to view templates stored by the metrics server **112** and/or the mapping engine **118**, and designate one or more templates as "active."

[0050] Further, an embodiment of the GUI provided by the UI module **420** allows the end-user to create new templates and modify existing templates by selecting and/or defining